REMARKS

The rejection of claims 1-10, 13-21 and 25 was maintained in the Advisory Action mailed on 07/25/2008 (hereafter "Outstanding Advisory Action"). Claims 1-2, 5-10, 13, 16-17, 20-21 and 25 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Gostanian *et al*, U.S. 5,781,910 (Gostanian), in view of "XA Components, Oracle8*i* JDBC Developer's Guide and Reference, 1999" (JDBC Guide).

By virtue of this paper, all the independent claims 1, 7, 10 and 16 are sought to be amended. The amendments are made without prejudice or disclaimer. The amendments are believed not to introduce new matter and their entry is respectfully requested. Claims 1-10, 13-21 and 25 are presented for consideration, further in view of below remarks.

Previous response Under 37 CFR § 1.116

It is respectfully requested that the previous response filed on 07/17/2008 under 37 CFR § 1.116 be entered and considered along with the present paper.

Response to Advisory Action

Without acquiescing to the Examiner's contentions, Applicants note how the presented claims address each point of the Outstanding Advisory Action.

Broadly, it is first pointed that the present amendment, "... wherein said user program contains groups of instructions to implement respective program logic for each of said task procedure and said rollback procedure, whereby each user program can have corresponding custom logic for the corresponding pair of task procedure and rollback procedure." in each of the independent claims, clearly addresses at least the material issues raised in the Outstanding Advisory Action.

In paragraph 1 of the Outstanding Advisory Action, it was stated that:

> ... does NOT place the application in condition for allowance because: As per Applicant's argument that Gostanian does not teach the claimed "user program," the Examiner respectfully disagrees. First, Applicant argues that the referenced application code 624 of Fig. 6 corresponds to the claimed "user program," but that it does not teach the claimed functionality, i.e. the ability to specify custom rollback and task procedures. ***The Examiner did not cite Fig. 6, however, the***

> ***Examiner cited Fig. 3, specifically the application programs of clients 302-308.***
> (Paragraph 1 of Outstanding Advisory Action, ***Emphasis Added***)

Applicants apologize for not being clear on why Figure 6 was explained in the previous response dated 07/17/2008. Irrespective, the concern is believed to be moot in view of the foregoing amendments. In particular, with reference to the application programs of client 302-308, Gostanian discloses:

> Associated with each database site 312, 314 is an application server 332, 334, respectively. As illustrated in FIG. 3, the application servers 332, 334 may be layered over their respective database sites 312, 314. The application servers 332, 334 coordinate the requested database transactions for the application clients 302-308. Specifically, ***the application clients 302-308 request the execution of transactions preferably by issuing application programming interface (API) calls via the communications network 310 to the application servers 332, 334.*** The application servers 332, 324 then communicate with each other as described below and with their corresponding database servers 312, 314, through conventional RDBMS API calls, such as ORACLE Pro*C.RTM. or SYBASE Embedded SQL/C.RTM. precompiler statements in order to update the relevant data items in response to the requested transactions.
> (Col. 9, lines 27-43 of Gostanian, ***Emphasis Added***)

From the above, the clients 302-308 merely use API calls, with the program logic for the implementation of the calls being provided in the application servers 332, 334.

At least in view of the foregoing amendment, the above concern is rendered moot.

The same applies to the second paragraph in the Outstanding Advisory Action, reproduced below for the convenience of the Examiner:

> Second, Applicant argues (with respect to Fig. 3) that the application servers 332, 324 and manager processes 336, 338 coordinate the commits and aborts. Applicant has not claimed, however, that it is the user program that coordinates commits and aborts. Rather, the user program requests a unique identifier, specifies combinations, and specifies task procedure order. ***Nothing in the claims precludes a server from coordinating transactions based on the specifications in the user program.***
> (Paragraph 2 of Outstanding Advisory Action, ***Emphasis Added***)

It is respectfully pointed out that foregoing amendments now expressly recite that the program logic for task/rollback procedures is provided in the user programs.

In paragraph 3 of the Outstanding Advisory Action, it was stated that:

> Gastanian [*sic*] does not teach specifying combinations in the user program, but the JDBC Guide does. Applicant correctly argues that a transaction manager uses the XA resource instances to coordinate all transaction branches. ***The cited XAResource interface is located in the oracle.jdbc.xa.client and server packages, however, indicating that its fields are used by both client (implementing the user program) and server, not just the server.***
> (Paragraph 3 of Outstanding Advisory Action, ***Emphasis Added***)

It is respectfully noted that XAResource is an 'interface' as admitted in the above-quoted text. An interface merely indicates that the clients have access to the fields (labels, names of the methods, etc.), but does not mean that the program logic is implemented within the clients/user programs.

Indeed, JDBC Guide expressly teaches that the program logic is a part of the resource class/driver on the transaction manager handling the transactions, based on the below:

> Oracle JDBC implements the XAResource interface with the OracleXAResource class, located both in the oracle.jdbc.xa.client package and the oracle. jdbc.xa.server package.
> ***The Oracle JDBC driver creates and returns an OracleXAResource instance whenever the OracieXAConnection class getxAResource () method is called, and it is the Oracle JDBC driver that associates an XA resource instance with a connection instance and the transaction branch being executed through that connection.***
> This is how an OracleXAResource instance is associated with a particular connection and with the transaction branch being executed in that connection.
> (Page 3 of JDBC Guide, ***Emphasis Added***)

Thus, as program logic of the task/rollback procedures are in the JDBCDriver, the logic of which is shared by the application programs (akin to the claimed user programs) of different clients, it is concluded that JDBC Guide also does not teach or reasonably suggest the claimed feature of the program logic (of task/rollback procedures) being provided **in the user programs**.

The above conclusion (see "transparent to application programs" below) is further supported by:

> A transaction manager, receiving OracleXAResource instances from a middle-tier component such as an application server, would typically invoke this functionality.

> ***Each of these methods takes a transaction ID as input, in the form of an Oraciexid instance***, which includes a transaction branch ID component and a distributed transaction ID component. Every transaction branch has a unique transaction ID, but transaction branches belonging to the same distributed transaction have the same distributed transaction component as part of their transaction Ids.
> (Page 4 paragraphs 4-5 of JDBC Guide, ***Emphasis Added***)
> ...
> An XA transaction ID instance is an instance of a class that implements the standard javax.transaction.xa.xid interface, which is a Java mapping of the X/Open transaction identifier XID structure.
> Oracle implements this interface with the OracleXid class in the oracle. jdbc. xa package. ***OracleXid instances would be used only in a transaction manager, transparent to application programs or an application server.***
> (Page 7, last 3 paragraphs of JDBC Guide, ***Emphasis Added***)

Accordingly, it is concluded that Gostanian and JDBC Guide, neither individually nor in combination, do not disclose or reasonably suggest one or more features of each currently amended independent claims.

The Examiner has further stated in the Outstanding Advisory Action:

> Lastly, the specification does not explicitly define "user program," thus the Examiner must apply the broadest reasonable interpretation to the term. There is nothing in the claims that require the user program to be implemented solely on a client device. Rather, ***it could just as reasonably be written by an administrator or programmer for a central transaction manager.***
> (Paragraph 4 of Outstanding Advisory Action, ***Emphasis Added***)

The above concern is also rendered moot in view of the foregoing amendments. In particular, program logic in the 'central transaction manager' would affect all the user programs using the transaction manager, thereby precluding the claimed ability for each user program to contain corresponding custom logic for task/rollback procedures.

Thus, all the currently amended independent claims are allowable over the art of record. The dependent claims are allowable at least for some of the reasons noted above with respect to the independent claims.

Conclusion

Thus, it is believed that all rejections have been overcome and the application is in condition for allowance. The Examiner is respectfully requested to continue examination and pass the case to issuance. The Examiner is invited to telephone the undersigned representative at 707.356.4172 if it is believed that an interview might be useful for any reason.

Respectfully submitted,

/Narendra Reddy Thappeta/
Signature
Printed Name: Narendra Reddy Thappeta
Attorney for Applicant
Registration Number: 41,416

Date: August 28, 2008